【原著論文】

辞書式最適ネットワークフローによる公平なクラス編成問題へのアプローチ

永野　清仁 1)　・　吉良　知文 2)

1) 数理情報学研究室
2) ソーシャル数理研究室

Lexicographically optimal network flow approaches to fair class assignment problems

Kiyohito NAGANO[1), 　Akifumi KIRA[2)]

1) Mathematical Informatics
2) Social Mathematics

Abstract

Network flow problems, which are a subclass of linear optimization problems, are not only computationally tractable, but also applicable to real-world problems in various fields. The class assignment problem is a research topic to which network flow algorithm can be applied. In this paper, we introduce the concepts of lexicographically optimal class assignments, and give network flow-based algorithms for the fair class assignment problems. In addition, we evaluate the performance of the proposed methods through computational experiments.

キーワード：数理最適化，線形最適化，線形計画，ネットワークフロー

1. はじめに

大学教育などにおいて，数十人から数千人規模の学生を，定員のある複数の授業（クラス）のどれかに配属させるような状況がしばしばある．このような問題はクラス編成問題として知られている．例えば群馬大学では，大学 1 年次の教養教育のある科目において，1000 名程度の学生を学部等の区別なく，幅広いトピックを扱う 50 程度の授業に振り分ける必要があり，授業への配属の最適性や公平性が望まれている．クラス編成問題は，学科やコース，研究室への割当などとも関連する基本的な問題である．クラス編成問題に対しては，これまで様々な手法が提案されている [4, 5, 8, 9, 10]．クラス編

成問題は，安定結婚問題 [3] と密接に関連している．安定結婚問題の手法としてよく知られているゲール・シャプレーのアルゴリズム [2] は受け入れ保留方式ともよばれ，クラス編成問題に応用されている．その他，受け入れ保留方式よりも単純な方法であるボストン方式や，トップ・トレーディング・サイクル方式などがクラス編成問題へのアプローチとして知られている [10].

本研究では，最適化モデルに基づく手法 [4, 8, 9] をベースにクラス編成問題を扱う．この手法を用いる際に解く数理最適化問題は，ネットワークフロー問題 [1, 6] として扱うことができる．ネットワークフロー問題として解く，あるいは線形計画問題として解くことで，通常のパソコンであってもそれなりに大きい規模，例えば学生数が数千程度の規模，の問題を非常に高速に解くことができる (プログラミング言語 Python を用いた数理最適化については [7] などを参照されたい)．最適化モデルに基づくクラス編成では,「学生の満足度」というパラメータを設定する点に注意が必要である．満足度を学生自身に個別に決めさせるアプローチ [4] なども提案されているが，本研究では学生の満足度は学生間で差がないという単純なモデルについて，適切な満足度のパラメータ設定をすることに重点を置く.

クラス編成問題に関して，本研究では,「辞書式最大なクラス編成」と「辞書式最小なクラス編成」という 2 つのアプローチを提案する．これらのクラス編成は，大まかに言って以下のような性質を満たす.

辞書式最大なクラス編成：

希望順位の高い授業にできるだけ多くの学生を配属させるようなクラス編成になっている.
言い換えれば，幸福な学生をなるべく増やすようなクラス編成である.

辞書式最小なクラス編成：

希望順位の低い授業にできるだけ学生を配属させないような編成になっている.
言い換えれば，不幸な学生をなるべく減らすようなクラス編成である.

これらをまとめて辞書式最適なクラス編成とよぶことにする．辞書式最適なクラス編成は，満足度というパラメータに左右されないクラス編成である．特に辞書式最小なクラス編成は，全体最適を目指すのではなく，偏りを小さくすることを目指す編成であるため，ある意味で「公平な」クラス編成とみなすことができる．本稿では，辞書式最適クラス編成問題が，ネットワークフロー問題として効率的に解けることを示す.

本研究で提案する辞書式最適なクラス編成そのものが必ずしも良いクラス編成になっているとは考えていない．満足度パラメータをある値に設定して，その満足度に関する数理最適化問題を解いてクラス編成を求める際に，最適解として出力されたクラス編成がどの程度良いかを考察するために，その判断材料として辞書式最適なクラス編成を用いることができる．辞書式最適なクラス編成のようにいわば極端な最適化の場合にはどうなるかという情報を参考にして，妥当なクラス編成を模索するのが良いであろう.

本論文の構成は以下の通りである．2 章では，クラス編成問題の設定と最適化モデルについて解説する．3 章では，最適化モデルを用いる場合，満足度の設定によりクラス編成の結果が大きく変わってくることを，具体例を用いて考察する．4 章では，辞書式最大なクラス編成，辞書式最小なクラス編成を導入し，それらネットワークフロー問題に帰着して求める方法について述べる．さらに，計算機実験により，提案手法の妥当性について検証する．

2. クラス編成問題の設定と最適化モデル

クラス編成問題は n 人の学生（n は 100 や 1000 など）を，それぞれ定員が定まっている m 個の授業（m は 10 や 50 など）のどれかに配属させる問題である．本章では，まず本研究で扱うクラス編成問題の問題設定について説明し, さらにこれを解くための標準的な最適化モデル [4, 8, 9] を説明する．

2.1. 問題設定

n 人の学生を m 個の授業に配属させるクラス編成問題について，本研究で扱う問題設定や，約束事は以下の通りである．

(i) i を学生の番号とし，j を授業の番号とする．i は 1 から n の整数値をとり，j は 1 から m の整数値をとる．

(ii) クラス編成において，各学生 $i \in \{1, \dots, n\}$ は，それぞれちょうど 1 つの授業に割り当てられる．つまり授業に配属されないことや，2 つ以上の授業に配属されることを認めない．

(iii) クラス編成において，各授業 $j \in \{1, \dots, m\}$ には，それぞれ定員 b_j (正の整数) が定まっている．授業 j に配属される学生の数は定員 b_j を超えない．さらに，全授業の定員の和は n 以上であるとする．つまり次式が成り立つ．

$$\sum_{j=1}^{m} b_j = b_1 + b_2 + \cdots + b_m \geq n$$

(iv) 各学生 $i \in \{1, \dots, n\}$ は，授業について第 1 希望から第 k 希望（k は学生によらない定数）までを決定し，申告しているものとする．第 1 希望等を複数にするというようなことは認めない．k の値として，$k = m$ であれば当然，全学生を第 k 希望以内の授業に必ず配属することができる．ただ，現実問題としては，k を大きくすると学生にとって負担が大きくなる．その一方で，k が小さいときは，今度は第 k 希望以内の授業に配属できない学生が出てきてしまうことに注意する．

(v) 学生の情報として (iv) の希望順位以外の情報（成績データ等）は利用しない．

これらの条件の下で，学生の希望を最大限考慮するようなクラス編成を求めることを考える．学生の成績を考慮して，成績補正を行うクラス編成 [9] なども考えられるが，本稿ではそのような仮定を置かずに議論する．例えば，大学 1 年生を対象とする授業の場合，成績情報を用いることが困難な場合

もある．

以下では，クラス編成問題の数理最適化としての定式化について説明していく．

2.2. 最適化モデル

この節では，クラス編成問題に対するアプローチとして，標準的な最適化モデル [4, 8, 9] について説明する．この最適化モデルは，本研究で提案する辞書式最適なクラス編成の基礎となる考え方である．まず，数理最適化の基本事項について整理し，その後にクラス編成問題の最適化モデルについて記述する．

2.2.1 数理最適化問題とは

数理最適化問題とは，多くの場合，次のように書くことができる問題のことである．

$$\begin{aligned}&\text{目的：}\quad f(\boldsymbol{x}) \ \rightarrow\ \text{最大 (または最小)}\\&\text{制約：}\quad \boldsymbol{x} \in S\end{aligned} \tag{1}$$

数理最適化問題の基本形である問題 (1) について，その詳細や補足説明を以下に記述する．

- $\boldsymbol{x}$ は各成分が実数値をとるベクトルであり，次元を d とする．$\boldsymbol{x}$ の各成分が整数値のみをとると限定する場合もある．$f(\boldsymbol{x})$ は実数値であり，f を問題 (1) の目的関数とよぶ.
- $\boldsymbol{x} = (x_1,\ x_2, \dots\ ,\ x_d)$ のうちで，制約の条件を満たすものを実行可能解とよび，実行可能解全体 S を実行可能領域とよぶ.
- 数理最適化問題 (1) は，実行可能解 $\boldsymbol{x} \in S$ の中で目的関数値 $f(\boldsymbol{x})$ が最大 (または最小) となるものを見つける問題である．問題 (1) において，目的関数を最大 (または最小) にする $\boldsymbol{x}$ を最適解とよび，そのときの目的関数値 $f(\boldsymbol{x})$ を最適値とよぶ.
- 以前は数理最適化 (mathematical optimization) ではなく, 数理計画 (mathematical programming) という呼び方が主流であった. 現在は,「数理計画」と「数理最適化」が両方使われているが, 徐々に数理最適化にシフトする流れがある.

数理最適化問題の重要なサブクラスとして, 線形計画問題, ネットワークフロー問題, 整数計画問題, 混合整数計画問題などがある.

線形計画問題（Linear Programming problem）とは，数理最適化問題 (1) の一種であり，各変数が実数値をとる連続変数，目的関数が 1 次式，実行可能領域が凸多面体 (いくつかの 1 次不等式により定まる領域の共通部分) として定まるもののことである. 線形計画問題は LP ともよばれることもある. LP の重要なサブクラスとして，最大流問題や最小費用流問題などのネットワークフロー問題 [1] がある．LP 自体が理論的には比較的扱いやすい数理最適化問題であるが，ネットワークフロー問題は, 解の整数性など，一般の LP よりもさらに良い性質を持つ.

LP は実数変数に関する最適化問題であるが，変数がすべて整数変数であるような数理最適化問題は整数計画問題（Integer Programming problem）とよばれ IP と略される．また，整数変数と実数変数が混ざった数理最適化問題は混合整数計画問題（Mixed Integer Programming problem）とよばれ MIP と略される．LP は多項式時間で解けることが知られている問題，つまり扱いやすいタイプの問題である．その一方で，一般的に IP と MIP はともに NP 困難な最適化問題であり，理論的には扱いやすい問題とは言えない. しかし, 近年ではアルゴリズムの発達とコンピュータの性能の向上によって, IP や MIP もそれなりのサイズの問題が数理最適化ソルバによって実時間で解けるようになってきている．実際にコンピュータを使った解き方については [7] などを参照されたい.

2.2.2 クラス編成問題の最適化モデル

2.1 節で述べたクラス編成問題を数理最適化問題 (1) の形で表すことで，クラス編成問題の標準的な最適化モデル [4, 8, 9] を導出する.

まず，クラス編成を数式で表現するために，学生 $i \in \{1, \dots, n\}$ を授業 $j \in \{1, \dots, m\}$ に配属させるかどうかを表す 0 または 1 をとる変数 (0-1 変数) x_{ij} を次のように定める.

$$x_{ij} = \begin{cases} 1, & \text{学生 } i \text{ を授業 } j \text{ に割り当てる} \\ 0, & \text{学生 } i \text{ を授業 } j \text{ に割り当てない} \end{cases} \tag{2}$$

式 (2) のような 0-1 変数 $x_{ij} \in \{0, 1\}$ は $n \times m$ 個ある. クラス編成を決定することは, すべての x_{ij} を決定することに対応する．以下では，nm 個の 0-1 変数 x_{ij} をまとめた nm 次元ベクトル $\boldsymbol{x}$ と表記し，nm 次元の 0-1 ベクトル $\boldsymbol{x} \in \{0,1\}^{nm}$ とクラス編成を同一視する.

続いて，クラス編成 $\boldsymbol{x} \in \{0,1\}^{nm}$ が満たすべき条件について説明する．学生 $i \in \{1, \dots, n\}$ が，それぞれちょうど 1 つの授業に配属されるには，次の条件が成立すればよい.

$$\sum_{j=1}^{m} x_{ij} = x_{i1} + x_{i2} + \cdots + x_{im} = 1, \qquad \forall i \in \{1, \dots, n\} \tag{3}$$

また，授業 $j \in \{1, \dots, m\}$ に配属される学生の数は定員 b_j を超えないという条件は，次式のように表すことができる.

$$\sum_{i=1}^{n} x_{ij} = x_{1j} + x_{2j} + \cdots + x_{nj} \leq b_j, \qquad \forall j \in \{1, \dots, m\} \tag{4}$$

ベクトル $\boldsymbol{x} \in \{0,1\}^{nm}$ が式 (3) と式 (4) をともに満たすとき, $\boldsymbol{x} \in \{0,1\}^{nm}$ をクラス編成問題の実行可能解，あるいは実行可能なクラス編成とよぶことにする.

目的関数を適切に設定することで，クラス編成問題を数理最適化問題として定式化することができ

る．クラス編成問題の目的関数を定めるために，満足度 (効用) を p_{ij} を次のように定める (p_{ij} の値をどのように定めるかは 2.2.3 項で議論する).

p_{ij}：学生 i が授業 j に配属されたときの学生 i の満足度 (効用)

このとき，実行可能なクラス編成 $\boldsymbol{x}$ について，学生 i の満足度は次のように表される．

$$(\text{学生 } i \text{ の満足度}) = \sum_{j=1}^{m} p_{ij}x_{ij} = p_{i1}x_{i1} + p_{i2}x_{i2} + \cdots + p_{im}x_{im}$$

よって，実行可能なクラス編成 $\boldsymbol{x}$ について，全学生の満足度の和は次式で定まる．

$$(\text{全学生の満足度の和}) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij} \tag{5}$$

以上で，クラス編成問題を数理最適化問題 (1) として定式化する準備が整った．実行可能なクラス編成 $\boldsymbol{x}$ の中で，全学生の満足度の和を最大にするものを求める問題は以下のように記述できる．

$$\begin{aligned}
\text{目的：}\quad & \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij} \rightarrow \text{ 最大} \\
\text{制約：}\quad & \sum_{j=1}^{m} x_{ij} = 1, \qquad \forall i \in \{1, \dots, n\} \\
& \sum_{i=1}^{mn} x_{ij} \le b_j, \qquad \forall j \in \{1, \dots, m\} \\
& x_{ij} \in \{0, 1\},\ \forall i \in \{1, \dots, n\};\ \forall j \in \{1, \dots, m\}
\end{aligned} \tag{6}$$

問題 (6) は各変数が整数であるため，整数計画問題の形をしている．特に，各変数が 0 か 1 の値をとるため，0-1 整数計画問題とよばれるタイプの問題である．ここで，各 x_{ij} に関する「$x_{ij} \in \{0, 1\}$」をすべて「$0 \le x_{ij} \le 1$」に緩和することで得られる次の問題を考える．

$$\begin{aligned}
\text{目的：}\quad & \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij} \rightarrow \text{ 最大} \\
\text{制約：}\quad & \sum_{j=1}^{m} x_{ij} = 1, \qquad \forall i \in \{1, \dots, n\} \\
& \sum_{i=1}^{mn} x_{ij} \le b_j, \qquad \forall j \in \{1, \dots, m\} \\
& 0 \le x_{ij} \le 1,\ \forall i \in \{1, \dots, n\};\ \forall j \in \{1, \dots, m\}
\end{aligned} \tag{7}$$

問題 (7) は各変数が実数であり，線形計画問題の形をしている．実は問題 (7) は線形計画問題の中でも，最小費用流問題とよばれるタイプのネットワークフロー問題 [1] になっている．ネットワークフ

ロー問題の持つ良い性質から，(各 b_j が整数値であるため) 問題 (7) の最適化として変数 x_{ij} がすべて整数であるようなものの存在性が導かれる．各変数 x_{ij} が $0 \le x_{ij} \le 1$ を満たすことから，問題 (7) の最適化 $\boldsymbol{x}$ として，$\boldsymbol{x} \in \{0,1\}^{nm}$ であるようなものが存在することになる．さらに，線形計画問題 (7) を数理最適化ソルバで解くことで，そのような $\boldsymbol{x} \in \{0,1\}^{nm}$，つまり整数計画問題 (6) の最適解が得られると思ってよい (もちろん，整数の解が得られたかは，チェックする必要がある).

以上の議論をまとめると，クラス編成問題は，0-1 整数計画問題 (6) として定式化できるが，この問題の最適解は，線形計画問題 (7) を解くことで，整数計画問題として解くよりも効率的に得ることができる，ということになる．

2.2.3 クラス編成問題における満足度の設定と希望の割当結果

最適化モデルに基づいたクラス編成問題 (6) や (7) において，学生数 n や授業数 m，各授業の定員 b_j は現実問題として自由に決めることはできないが，満足度 p_{ij} はこちらで設定することが可能である．また，満足度を設定することで最適解であるクラス編成 $\boldsymbol{x}$ が決定するが，得られたクラス編成について，どの程度学生の希望通りの割当になっているかを検討する必要がある．ここではまず，学生の満足度 p_{ij} の設定について述べ，続いてクラス編成 $\boldsymbol{x}$ がどの程度良いかを検討するために，希望割当ベクトルの概念を導入する．

学生の満足度 p_{ij} の設定について述べる．本稿では満足度について学生を区別しないものとする．授業数 m が 10 であり，各学生が第 1～4 希望を申告する，つまり $k = 4$ の場合には，例えば次のような形で満足度を設定する．

満足度の設定方法の例

- 第 1 希望のクラス j に割り当てられたとき，その学生 i の満足度 p_{ij} は 100,
- 第 2 希望のクラス j に割り当てられたとき，その学生 i の満足度 p_{ij} は 70,
- 第 3 希望のクラス j に割り当てられたとき，その学生 i の満足度 p_{ij} は 30,
- 第 4 希望のクラス j に割り当てられたとき，その学生 i の満足度 p_{ij} は 0,
- 第 5～10 希望のクラス j に割り当てられたとき，その学生 i の満足度 p_{ij} は -1000

一般には，$a_1 \ge a_2 \ge \cdots \ge a_m$ として，クラス編成問題 (6) や (7) における満足度 p_{ij} を次のように定めることを，最適化モデルの基本とする．

$$p_{ij} = \begin{cases} a_1, & \text{学生 } i \text{ がクラス } j \text{ を第 1 希望としている} \\ a_2, & \text{学生 } i \text{ がクラス } j \text{ を第 2 希望としている} \\ \vdots & \qquad\qquad \vdots \\ a_m, & \text{学生 } i \text{ がクラス } j \text{ を第} m \text{希望としている} \end{cases}$$

$\boldsymbol{a} = (a_1, a_2, \dots, a_m)$ を満足度ベクトルとよぶことにする．最適化モデルにおいて，満足度ベクトルをどのように設定するかが重要となる．

続いて，実行可能なクラス編成 $\boldsymbol{x}$ に対し，$\boldsymbol{x}$ がどの程度学生の希望通りの割当になっているかを表すようなベクトル $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ を次で定める．

z_1: クラス編成 $\boldsymbol{x}$ で第 1 希望のクラスに割り当てられる学生の数

z_2: クラス編成 $\boldsymbol{x}$ で第 2 希望のクラスに割り当てられる学生の数

$\vdots$

z_m: クラス編成 $\boldsymbol{x}$ で第 m 希望のクラスに割り当てられる学生の数

以下では，$\boldsymbol{z}$ を $\boldsymbol{x}$ に関する希望割当ベクトルとよぶことにする．２つのクラス編成 $\boldsymbol{x}^{(1)}$ と $\boldsymbol{x}^{(2)}$ があったとき，それぞれの希望割当ベクトルを比較することで，クラス編成の「良さ」を比較することは自然である．意味を考えれば，$\boldsymbol{a}$ と $\boldsymbol{z}$ の内積 $\langle \boldsymbol{a}, \boldsymbol{z} \rangle = a_1 z_1 + a_2 z_2 + \cdots + a_m z_m$ は，クラス編成問題 (6) の目的関数の値と一致する．

3 章では，満足度ベクトルの設定によって，最適化モデルにより得られるクラス編成の希望割当ベクトルが大きく変わることを観察する．

3. クラス編成問題の例と最適化モデル

本章では，最も良いクラス編成がどのようなものかを決めることは，2 章で解説した最適化モデルを用いたとしても困難であることを，具体例を通じて理解する．

3.1 クラス編成問題とそれに対するクラス編成の例

表 1 の状況のクラス編成問題を考える．つまり，学生数 $n = 5$, 授業数 $m = 5$ とし，各授業の定員がすべて 1 であるようなクラス編成問題を考える．

表 1. 学生のクラス希望状況の例（学生数 5，クラス数 5）

学生	C1 (定員 1) 希望順位	C2 (定員 1) 希望順位	C3 (定員 1) 希望順位	C4 (定員 1) 希望順位	C5 (定員 1) 希望順位
S1	1	2	3	4	5
S2	1	2	3	4	5
S3	5	1	2	3	4
S4	4	5	1	2	3
S5	3	4	5	1	2

この問題について，まずは妥当なクラス編成にはどのようなものがあるかを考察し，さらに得られたクラス編成について，2.2.3 項で定義した希望割当ベクトルを比較する．

クラス編成問題の例題 (表 1) について，まず次の戦略を考えるのは自然である．

戦略 1： 第 1 希望に割り当てられる学生数を最大にするようなクラス編成を求める．

この戦略 1 に基づいて得られるクラス編成 $\boldsymbol{x}^{(1)}$ は表 2 の通りである．クラス編成 $\boldsymbol{x}^{(1)}$ は，学生 S1 を第 1 希望の C1 に，学生 S2 を第 5 希望の C5 に，学生 S3 を第 1 希望の C2 に，学生 S4 を第 1 希望の C3 に，学生 S5 を第 1 希望の C4 にそれぞれ配属させるようなものになっている．クラス編成$\boldsymbol{x}^{(1)}$について，第 1 希望の学生 4 名，第 2〜4 希望の学生がそれぞれ 0 名，第 5 希望の学生が 1 名であることから，希望割当ベクトル (2.2.3 項) $\boldsymbol{z}^{(1)}$ は $(4, 0, 0, 0, 1)$ となる．

表 2. 問題例 (表 1) に対するクラス編成 $\boldsymbol{x}^{(1)}$

学生	C1	C2	C3	C4	C5
S1	①	2	3	4	5
S2	1	2	3	4	⑤
S3	5	①	2	3	4
S4	4	5	①	2	3
S5	3	4	5	①	2

続いて，単純に第 1 希望を増やすのではなく，第 5 希望の授業に配属される学生がいないようにしつつ，第 1 希望に配属される学生を増やすような，次の戦略を考える．

戦略 2： すべての学生が第 4 希望以内のクラスに割り当てられるようにしつつ，第 1 希望に割り当てられる学生を最大にするクラス編成を求める．

この戦略 2 に基づいて得られるクラス編成の 1 つとしては表 3 の $\boldsymbol{x}^{(2)}$ があり，希望割当ベクトル $\boldsymbol{z}^{(2)}$ は $(3, 1, 0, 1, 0)$ となる．

表 3. 問題例 (表 1) に対するクラス編成 $\boldsymbol{x}^{(2)}$

学生	C1	C2	C3	C4	C5
S1	①	2	3	4	5
S2	1	2	3	④	5
S3	5	①	2	3	4
S4	4	5	①	2	3
S5	3	4	5	1	②

戦略 2 ではすべての学生が第 1〜4 希望の授業に配属されるようにしたが，第 1〜3 希望に限定する，あるいは第 1, 2 希望に限定することによって，それぞれ次の戦略が得られる．

戦略 3：　すべての学生が第 3 希望以内のクラスに割り当てられるようにしつつ，第 1 希望に割り当てられる学生を最大にするクラス編成を求める．

戦略 4：　すべての学生が第 2 希望以内のクラスに割り当てられるようにしつつ，第 1 希望に割り当てられる学生を最大にするクラス編成を求める．

戦略 3 により得られるクラス編成は表 4 の $\boldsymbol{x}^{(3)}$ であり，希望割当ベクトル $\boldsymbol{z}^{(3)}$ は $(3,0,2,0,0)$ となる．ちなみに $\boldsymbol{x}^{(3)}$ は，戦略 2 から得られるクラス編成の 1 つにもなっている．戦略 4 から得られるクラス編成は表 5 の $\boldsymbol{x}^{(4)}$ であり，希望割当ベクトル $\boldsymbol{z}^{(4)}$ は $(1,4,0,0,0)$，となる．

表 4. 問題例 (表 1) に対するクラス編成 $\boldsymbol{x}^{(3)}$

学生	C1	C2	C3	C4	C5
S1	①	2	3	4	5
S2	1	2	③	4	5
S3	5	①	2	3	4
S4	4	5	1	2	③
S5	3	4	5	①	2

表 5. 問題例 (表 1) に対するクラス編成 $\boldsymbol{x}^{(4)}$

学生	C1	C2	C3	C4	C5
S1	①	2	3	4	5
S2	1	②	3	4	5
S3	5	1	②	3	4
S4	4	5	1	②	3
S5	3	4	5	1	②

最後に，全体の人数のバランスを考慮するような，次の戦略を考える．

戦略 5：　すべての学生が第 3 希望以内のクラスに割り当てられるようにしつつ，第 1 から第 3 のバランスが極端にならないように配慮する．

戦略5により得られるクラス編成は表6の $\boldsymbol{x}^{(5)}$ であり, 希望割当ベクトル $\boldsymbol{z}^{(5)}$ は $(2,2,1,0,0)$ となる．

表 6. 問題例 (表 1) に対するクラス編成 $\boldsymbol{x}^{(5)}$

学生	C1	C2	C3	C4	C5
S1	①	2	3	4	5
S2	1	2	③	4	5
S3	5	①	2	3	4
S4	4	5	1	②	3
S5	3	4	5	1	②

以上の 5 つ戦略 1〜5 に基づいて得られるクラス編成 $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(5)}$ と，それらに対応する希望割当ベクトル $\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(5)}$ は，それぞれ以下のようになる．

- 戦略 1 に基づくクラス編成は $\boldsymbol{x}^{(1)}$，希望割当ベクトルは $\boldsymbol{z}^{(1)}=(4,0,0,0,1)$
- 戦略 2 に基づくクラス編成は $\boldsymbol{x}^{(2)}$，希望割当ベクトルは $\boldsymbol{z}^{(2)}=(3,1,0,1,0)$

- 戦略 3 に基づくクラス編成は $\boldsymbol{x}^{(3)}$，希望割当ベクトルは $\boldsymbol{z}^{(3)} = (3,0,2,0,0)$
- 戦略 4 に基づくクラス編成は $\boldsymbol{x}^{(4)}$，希望割当ベクトルは $\boldsymbol{z}^{(4)} = (1,4,0,0,0)$
- 戦略 5 に基づくクラス編成は $\boldsymbol{x}^{(5)}$，希望割当ベクトルは $\boldsymbol{z}^{(5)} = (2,2,1,0,0)$

希望割当ベクトル $\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(5)}$ を眺めたとき，考慮しているクラス編成の性質は互いに大きく異なっていることがわかる．そして $\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(5)}$ は，ある意味では，すべて妥当なクラス編成であり，これらに対して単純な優劣をつけることは困難である．

例えば, $\boldsymbol{z}^{(3)}$ と $\boldsymbol{z}^{(5)}$ の比較を考えてみる. 満足度ベクトルを $\boldsymbol{a} = (100, 90, 70,\ 0,\ 0)$ とした場合は, クラス編成問題 (6) の目的関数値は内積 $\langle \boldsymbol{a}, \boldsymbol{z} \rangle$ に一致することを思い出せば, $\langle \boldsymbol{a}, \boldsymbol{z}^{(3)} \rangle < \langle \boldsymbol{a}, \boldsymbol{z}^{(5)} \rangle$ が成立することから，$\boldsymbol{z}^{(5)}$ の方が良いことになる．その一方で，$\boldsymbol{a} = (100, 80, 70,\ 0,\ 0)$ とした場合は, $\langle \boldsymbol{a}, \boldsymbol{z}^{(3)} \rangle > \langle \boldsymbol{a}, \boldsymbol{z}^{(5)} \rangle$ が成立するので, $\boldsymbol{z}^{(3)}$ の方が良いことになる. 結果的に $\boldsymbol{z}^{(3)}$ と $\boldsymbol{z}^{(5)}$ の優劣をつけることができない．このような関係が，$\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(5)}$ の任意の 2 つについて成立する.

3.2 満足度ベクトルと最適化モデルに基づくクラス編成の関係

2.2 節で記述した最適化モデルでは, 満足度ベクトル $\boldsymbol{a}$ を決めれば, 最適化問題 (7) を解くことで, 最適なクラス編成 $\boldsymbol{x}$ が求まり，その結果として希望割当ベクトル $\boldsymbol{z}$ が定まる．ここでは，$\boldsymbol{a}$ を変えたときに，$\boldsymbol{z}$ がどのように変わるかを見るために計算機実験を行う．表 1 の問題例に対し，満足度ベクトル $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5)$ $(a_1 \geq a_2 \geq a_3 \geq a_4 \geq a_5)$ をランダムにとり，希望割当ベクトル $\boldsymbol{z}$ を求めるということを 10000 回実行し，$\boldsymbol{z}$ の分布を眺める.

3.2.1 満足度ベクトルとクラス編成の関係をみるための計算機実験

計算機実験の詳細について述べる．満足度ベクトル $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5)$ は独立な 5 つの 0 以上 1 以下をとる標準一様 (擬似) 乱数 r_1, r_2, r_3, r_4, r_5 を用いて，以下のように設定した.

$$a_1 = r_1 + r_2 + r_3 + r_4 + r_5, \quad a_2 = r_1 + r_2 + r_3 + r_4,$$
$$a_3 = r_1 + r_2 + r_3, \quad a_4 = r_1 + r_2, \quad a_5 = r_1$$

このようにして得られる満足度ベクトル $\boldsymbol{a}$ と表 1 のクラス編成問題の例題について，対応する最適化問題 (7) を解いて希望割当ベクトル $\boldsymbol{z}$ を求めるということを 10000 回繰り返して, $\boldsymbol{z}$ の分布を調べた. 最適化問題 (7) を解くには，Python 3.7 において PuLP パッケージを利用し，ソルバは PuLP に付属している CBC ソルバを用いた．計算機は Intel(R) Core(TM) i7 (2.70GHz)，16GB のメモリのものを用いた．ただし，今回は問題のサイズが小さく，1 回問題を解くのにかかる時間は 0.1 秒未満であった.

3.2.2 最適化モデルに基づくクラス編成について

計算機実験の結果として得られた希望割当ベクトル $\boldsymbol{z}$ の分布は，表 7 の通りであった．同じ設定の

クラス編成問題であっても，満足度ベクトルが変わることで，得られるクラス編成の性質が大きく変わり得ることが見て取れる．つまり，満足度ベクトル $\boldsymbol{a}$ をどのように設定するかが，クラス編成問題においていかに重要であるかがわかる．また，希望割当ベクトルとして，$\boldsymbol{z}^{(5)} = (2,2,1,0,0)$ が 1 度も得られていないが，これは $\boldsymbol{z}^{(5)}$ が $\boldsymbol{z}^{(3)}$ と $\boldsymbol{z}^{(4)}$ の中点，つまり $\boldsymbol{z}^{(5)} = \frac{1}{2}\left(\boldsymbol{z}^{(3)} + \boldsymbol{z}^{(4)}\right)$ が成り立つため，線形な目的関数を扱う限りにおいては，$\boldsymbol{z}^{(5)}$ よりも $\boldsymbol{z}^{(3)}$ または $\boldsymbol{z}^{(4)}$ の方が良い解になってしまうためである (厳密に言えば，$\boldsymbol{z}^{(3)}$，$\boldsymbol{z}^{(4)}$，$\boldsymbol{z}^{(5)}$ で目的関数値が等しくなる可能性もある).

表 7. 希望割当ベクトル $\boldsymbol{z}$ の分布 ($\boldsymbol{z}$ を 10000 回計算)

希望割当ベクトル $\boldsymbol{z}$	割合
$\boldsymbol{z}^{(1)} = (4,0,0,0,1)$	31.82 %
$\boldsymbol{z}^{(2)} = (3,1,0,1,0)$	7.16 %
$\boldsymbol{z}^{(3)} = (3,0,2,0,0)$	22.74 %
$\boldsymbol{z}^{(4)} = (1,4,0,0,0)$	38.28 %
$\boldsymbol{z}^{(5)} = (2,2,1,0,0)$	0.00%
その他の希望割当ベクトル	0.00%

4. 辞書式最適なクラス編成

3 章では，満足度ベクトルのパラメータの設定の仕方によって，最適化モデルで得られるクラス編成もまた大きく変わることを観察した．本章では，パラメータの設定に依存しないクラス編成の方法である「辞書式最大なクラス編成」と「辞書式最小なクラス編成」を提案する．これらをまとめて，辞書式最適なクラス編成とよぶことにする．さらに，その計算方法を示し，人工データに対して計算機実験を行う．

4.1 辞書式最大なクラス編成

辞書式最大なクラス編成の考え方を説明する．クラス編成問題においては，実行可能なクラス編成 $\boldsymbol{x}$ に対し，希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ が定義されるが，この $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ を辞書式に最大化するようなクラス編成が辞書式最大なクラス編成である．辞書式最大なクラス編成の希望割当ベクトル $\boldsymbol{z}$ は，以下のようにして定まる．

Step 1:　実行可能なクラス編成が存在するという条件下で z_1 を最大にし，z_1 を固定.

Step 2:　実行可能なクラス編成が存在するという条件下で z_2 を最大にし，z_2 を固定.

⋮

Step *m*:　実行可能なクラス編成が存在するという条件下で z_m を最大にし，z_m を固定.

このようにして，$\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ が決定するが，この希望割当ベクトル $\boldsymbol{z}$ に対応するクラス編成が辞書式最大なクラス編成である．辞書式最大なクラス編成は，希望順位の高い授業にできるだけ多く

の学生を配属させることを目指すようなクラス編成になっている.

3 章の表 1 のクラス編成問題の例の場合，希望割当ベクトル $\boldsymbol{z}^{(1)} = (4, 0, 0, 0, 1)$ に対応するのが，辞書式最大なクラス編成である．第 1 希望を増やすことを優先することで，第 5 希望に配属される不幸な学生が出てきていることがわかる.

4.2 辞書式最小なクラス編成

続いて，辞書式最小なクラス編成の考え方を説明する．希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, \dots, z_m)$ について，辞書式最大の場合とは逆向きで，$z_m, z_{m-1}, \dots, z_1$ の順に辞書式に最小化していくのが辞書式最小なクラス編成のアイデアである．辞書式最小なクラス編成の $\boldsymbol{z}$ は，以下のようにして定まる.

Step 1:　実行可能なクラス編成が存在するという条件下で z_m を最小にし，z_m を固定.

Step 2:　実行可能なクラス編成が存在するという条件下で z_{m-1} を最小にし，z_{m-1} を固定.

⋮

Step *m*:　実行可能なクラス編成が存在するという条件下で z_1 を最小にし，z_1 を固定.

このようにして，最終的には希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, \dots, z_m)$ が決定し，これに対応するクラス編成が辞書式最小なクラス編成である．辞書式最小なクラス編成は，希望順位の低い授業配属される学生をできるだけ少なくすることを目指すクラス編成になっているといえる.

3 章の表 1 のクラス編成問題の例の場合，希望割当ベクトル $\boldsymbol{z}^{(1)} = (1, 4, 0, 0, 0)$ に対応するのが，辞書式最小なクラス編成である．希望順位の低い学生を減らすことを優先することで，第 1 希望に配属される学生の数が少なくなっていることがわかる.

4.3 辞書式最適なクラス編成を求める

辞書式最大なクラス編成と辞書式最小なクラス編成は，最適化モデルに基づくクラス編成最適化問題 (6) に帰着可能であり，つまりネットワークフロー問題に帰着させて解くことができる.

4.3.1 辞書式最大なクラス編成を求める

辞書式最大なクラス編成を求めるには，満足度ベクトル $\boldsymbol{a} = (a_1, a_2, \dots, a_m)$ を

$$\begin{cases} a_1 = & (n+1)^{m-1} \\ a_2 = & (n+1)^{m-2} \\ & \vdots \\ a_m = & (n+1)^0 \ (=1) \end{cases} \tag{8}$$

と設定して，クラス編成の最適化問題 (6) を解けばよい．なぜならば，この式 (8) の設定の下では，希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, \dots, z_m)$ において，まず z_1 の値を 1 つでも増やすことが優先され，続いて z_2 の値を 1 つでも増やすことが優先され，以下同様にして $\boldsymbol{z}$ が決まる．これはまさに，辞書式最大

なクラス編成そのものである．よって，式 (8) の満足度の設定で問題 (6) を解くことで，辞書式最大なクラス編成が得られる．

4.3.2 辞書式最小なクラス編成を求める

辞書式最小な場合も，辞書式最大の場合と考え方は同じである．満足度ベクトル $\boldsymbol{a}$ を

$$\begin{cases} a_1 = & -(n+1)^0 \ (= -1) \\ a_2 = & -(n+1) \\ & \vdots \\ a_m = & -(n+1)^{m-1} \end{cases} \tag{9}$$

と設定して，クラス編成の最適化問題 (6) を解けばよい．なぜならば，式 (9) の設定の下では，希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ において，まず z_m の値を 1 つでも減らすことが優先され，続いて z_{m-1} の値を 1 つでも減らすことが優先され，以下同様にして $\boldsymbol{z}$ が決まる．これはまさに，辞書式最小なクラス編成そのものである．

4.4 辞書式最適なクラス編成の性能評価

ここでは，2 章で扱った最適化モデルに基づくクラス編成手法と，本章で提案した辞書式最適なクラス編成手法について，性能を比較する計算機実験を行う．

計算機実験で用いるクラス編成問題の例題は次の通りとする．

<u>**計算機実験で用いるクラス編成問題**</u>

学生数 $n = 200$，授業数 $m = 5$

授業 1～5 の定員を，それぞれ 10, 25, 40, 55, 70 (合計が $n = 200$) と固定

n 人の学生の授業に対する希望順位はランダムに生成

この条件を満たす問題例を 100 問作成 (異なるのは n 人の学生の授業に対する希望順位のみ) して，次の 3 つの手法について同じ問題を解いて，希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, z_3, z_4, z_5)$ の比較をする．

(1) 最適化モデルに基づくクラス編成手法 (満足度ベクトル $\boldsymbol{a} = (100,\ 70,\ 40,\ 0, -1000)$)

(2) 辞書式最大なクラス編成手法

(3) 辞書式最小なクラス編成手法.

計算機環境については 3.2 節と同じで，最適化問題を解くのに Python 3.7 の PuLP パッケージと CBC ソルバを利用し，Intel(R) Core(TM) i7 (2.70GHz)，メモリ 16GB の計算機を用いた．また，問題を 1 問解くのにかかる時間は，手法 (1), (2), (3) でほとんど差がなく，すべて 1 秒未満であった.

手法 (1), (2), (3) について，希望割当ベクトル $\boldsymbol{z} = (z_1, z_2, z_3, z_4, z_5)$ の 100 個の平均と，100 個のうちで第 k 希望までにちょうどおさまった問題数 ($k = 2, 3, 4$) はそれぞれ以下のようになった.

手法(1) [最適化モデル]	($\boldsymbol{z}$ の平均) = (148.6, 46.9, 4.5, 0, 0)	
	第 2 希望までにちょうどおさまった問題数：	7
	第 3 希望までにちょうどおさまった問題数：	93
	第 4 希望までにちょうどおさまった問題数：	0
手法(2) [辞書式最大]	($\boldsymbol{z}$ の平均) = (154.2, 35.8, 8.8, 1.3, 0)	
	第 2 希望までにちょうどおさまった問題数：	1
	第 3 希望までにちょうどおさまった問題数：	58
	第 4 希望までにちょうどおさまった問題数：	41
手法(3) [辞書式最小]	($\boldsymbol{z}$ の平均) = (143.2, 56.7, 0.1, 0, 0)	
	第 2 希望までにちょうどおさまった問題数：	98
	第 3 希望までにちょうどおさまった問題数：	2
	第 4 希望までにちょうどおさまった問題数：	0

辞書式最大なクラス編成は，確かに第 1 希望に配属される学生数 z_1 の値は大きくなるが，その反面第 4 希望に配属される学生も出てくることがあるとわかる．また，辞書式最小なクラス編成は，ほとんどの場合で第 2 希望までにおさまるが，第 1 希望に配属される学生数がやはり少なくなる．

辞書式最適なクラス編成は，それ自体が優れた手法ということではない．辞書式最大・辞書式最小なクラス編成は「両極端な場合のクラス編成」であり，これらの情報を参考にして，最適化モデルに基づく妥当なクラス編成を作成するための助けになることを期待する．

5. おわりに

本研究では，最適化モデルに基づくクラス編成をベースにして辞書式最大なクラス編成と辞書式最小なクラス編成を提案し，これらのクラス編成最適化問題がネットワークフロー問題に帰着して解けることを示した．今回提案するアプローチや実験結果は基礎的なものであり，実データに適用した場合の性能評価など，様々な問題が残されている．

参考文献

[1] R. K. Ahuja, T. L Magnanti, and J. B. Orlin: “Network Flows: Theory, Algorithms, and Applications,” Prentice Hall, 1993.

[2] D. Gale and L. S. Shapley: “College admissions and the stability of marriage,” American Mathematical Monthly, Vol.69 (1962), pp. 9–15.

[3] D. Gusfield and R. W. Irving: “The Stable Marriage Problem: Structure and Algorithms,” The MIT Press, 1989.

[4] 今野浩：「数理決定法入門 キャンパスの OR」, 朝倉書店, 1992.

[5] 片岡達, 茨木俊秀：「研究室配属のための一方式の提案とその数理的考察」, Transactions of the

Operatiens Research Society of Japan, Vol.51 (2008), pp. 71−93.

[6] 久保幹雄, 松井知己, 田村明久(編集) :「応用数理計画ハンドブック」, 朝倉書店, 2002.

[7] 久保幹雄, 小林和博, 斉藤努, 並木誠, 橋本英樹:「Python 言語によるビジネスアナリティクス 実務家のための最適化・統計解析・機械学習」, 近代科学社, 2016.

[8] 堀田敬介: 「学生満足度の観点によるゼミ配属法の定量的比較」, 文教大学情報学部 情報研究, Vol.35 (2006), pp.367-378.

[9] 堀田敬介: 「最適化技術のクラス編成問題への適用」, 文教大学経営学部 経営論集, Vol.2 (2016), pp.1-18.

[10] 安田洋祐 :「学校選択制のデザイン ゲーム理論アプローチ」, NTT 出版, 2010.

原稿受領日　2019 年 9 月 6 日
修正原稿受領日　2019 年 11 月 1 日